



NRL/MR/7440--05-8867

A Faster Technique for the Transformation of Universal Transverse Mercator Projected Raster Images into a Geodetic Projection

HILLARY C. MESICK

ELIAS Z. K. IOUP

JOHN T. SAMPLE

*Mapping, Charting, and Geodesy Branch
Marine Geosciences Division*

March 11, 2005

20050329 012

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 11-03-2005		2. REPORT TYPE Memorandum Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE A Faster Technique for the Transformation of Universal Transverse Mercator Projected Raster Images into a Geodetic Projection				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Hillary C. Mesick, Elias Z. K. Ioup, and John T. Sample				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 74-8100-05	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Marine Geosciences Division Stennis Space Center, MS 39529-5004				8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/7440--05-8867	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Guard Bureau Counterdrug Office 925 Dalney Street Atlanta, GA 30332				10. SPONSOR / MONITOR'S ACRONYM(S)	
				11. SPONSOR / MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Often it is necessary to transform a georeferenced raster image from one projection to another. This report describes an efficient method for performing Universal Transverse Mercator (UTM) to Geodetic projection transformations. The reprojected images are suitable for applications that require high accuracy and can be seamlessly combined with preexisting Geodetic data. The procedure can be abstracted to convert between any two projection types.					
15. SUBJECT TERMS Raster image; Projection transformations					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 16	19a. NAME OF RESPONSIBLE PERSON Hillary Mesick
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code) (228) 688-5257

CONTENTS

INTRODUCTION.....	1
BACKGROUND.....	1
POINT WISE REPROJECTION	3
REPROJECTION PERFORMANCE	5
PRECOMPUTED TABLE.....	5
PROCESSING WITH FIDUCIALS	6
RESULTS.....	7
CONCLUSIONS.....	11
ACKNOWLEDGEMENTS	12
REFERENCES	13

A FASTER TECHNIQUE FOR THE TRANSFORMATION OF UNIVERSAL TRANSVERSE MERCATOR PROJECTED RASTER IMAGES INTO A GEODETIC PROJECTION

INTRODUCTION

Much of the source material for cartographic applications is frequently made available in the Universal Transverse Mercator (UTM) Projection. The U.S. Geological Survey currently supplies Digital Orthographic Photo Quads (DOQ) and Topographic Quads in this projection (USGS, 1996).

The goal of this project is to combine cartographic materials from differing projections into a composite map in Geodetic projection. In particular, DOQs in the UTM projection could be combined with vector roads and water features in Geodetic. Since Geodetic is a common projection format, easily displayed in many GIS packages, the UTM images were transformed into Geodetic for our application.

Unlike vector data where all the cartographic features are composed of explicit coordinates, raster or image data is composed of individual pixels with implicit coordinates determined by the projection employed at the creation of the image and the corner coordinates or other geo-referencing information. To transform vector data only the coordinates of the features need be manipulated. However, raster or image data requires that every pixel in the image be remapped.

As an example, consider the state of Georgia. A nominal DOQ is approximately 7000 by 7000 (49 million) pixels. Each of the 49 million pixels in a DOQ image must be remapped to a new coordinate system. The state of Georgia is covered by approximately 4000 DOQs, so there are 196 Billion transformations. Since the UTM to Geodetic transformation equations are numerically complex and must be performed a large number of times for each image, an efficient method was required to perform this transformation.

BACKGROUND

A necessary prerequisite for transforming images between projections is the ability to transform point data between projections. Projection transformations are defined in a series of equations. The specific equations for UTM to Geodetic transformations are actually the equations for Transverse Mercator to Geodetic transformations where constants specific to the UTM projection are used. Figures 1 and 2 give the appropriate forward and reverse transformations for UTM to Geodetic as given by Snyder (1987). An examination of the equations shows that they are computationally intensive. Any process which uses those equations extensively could suffer from serious performance problems. However, these point transformations are required regardless of the method used to transform images. An efficient method of using them is therefore a worthy pursuit.

$$\begin{aligned}
x &= k_0 N \left[A + (1 - T + C) A^3 / 6 + (5 - 18T + T^2 + 72C - 58e^2) A^5 / 120 \right] \\
y &= k_0 \left\{ M - M_0 + N \tan(\phi) \left[A^2 / 2 + (5 - T + 9C + 4C^2) A^4 / 24 + (61 - 58T + T^2 + 600C - 330e^2) A^6 / 720 \right] \right\} \\
k &= k_0 \left[1 + (1 + C) A^2 / 2 + (5 - 4T + 42C + 13C^2 - 28e^2) A^4 / 24 + (61 - 148T + 16T^2) A^6 / 720 \right] \\
\text{where} \\
k_0 &= \text{scale of central meridian (.9996 for UTM)} \\
e^2 &= e^2 / (1 - e^2) \\
N &= a / (1 - e^2 \sin^2 \phi)^{1/2} \\
T &= \tan^2 \phi \\
C &= e^2 \cos^2 \phi \\
A &= (\lambda - \lambda_0) \cos \phi \quad // \lambda, \lambda_0 \in \text{radians} \\
M &= a \left[(1 - e^2 / 4 - 3e^4 / 64 - 5e^6 / 256 - \dots) \phi - (3e^2 / 8 + 3e^4 / 32 + 45e^6 / 1024 + \dots) \sin 2\phi \right. \\
&\quad \left. + (15e^4 / 256 + 45e^6 / 1024 + \dots) \sin 4\phi - (35e^6 / 3072 + \dots) \sin 6\phi + \dots \right] \\
M_0 &= M \text{ calculated for } \phi_0, \text{ i.e. central meridian} \\
e &= 0.0818192180 \quad // \text{eccentricity for WGS84 ellipsoid} \\
a &= 6378137.0 \quad // \text{semi major axis of WGS84 ellipsoid} \\
\lambda &= \text{geodetic longitude} \quad // \text{radians} \\
\phi &= \text{geodetic latitude} \quad // \text{radians}
\end{aligned}$$

Figure 1: The UTM to Geodetic transformation formulas

$$\begin{aligned}
\phi &= \phi_1 - (N_1 \tan \phi_1 / R_1) \left[D^2 / 2 - (5 + 3T_1 + 10C_1^2 - 9e_1^2) D^4 / 24 + \right. \\
&\quad \left. (61 + 90T_1 + 298C_1 + 45T_1^2 - 252e_1^2 - 3C_1^2) D^6 / 720 \right] \\
\lambda &= \lambda_0 + [D - (1 + 2T_1 + C_1) D^3 / 6 + (5 - 2C_1 + 28T_1 - 3C_1^2 + 8e_1^2 + 24T_1^2) D^5 / 120] / \cos \phi_1 \\
\phi_1 &= \mu + (3e_1 / 2 - 27e_1^3 / 32 + \dots) \sin 2\mu + (21e_1^2 / 16 - 55e_1^4 / 32 + \dots) \sin 4\mu \\
&\quad + (151e_1^3 / 96 + \dots) \sin 6\mu + (1097e_1^4 / 512 - \dots) \sin 8\mu \\
e_1 &= [1 - (1 - e^2)^{1/2}] / [(1 - e^2)^{1/2}] \\
\mu &= M / [a(1 - e^2 / 4 - 3e^4 / 64 - 5e^6 / 256 - \dots)] \\
M &= M_0 + y / k_0 \\
e_1^2 &= e^2 / (1 - e^2) \\
C_1 &= e_1^2 \cos^2 \phi_1 \\
T_1 &= \tan^2 \phi_1 \\
N_1 &= a / (1 - e^2 \sin^2 \phi_1)^{1/2} \\
R_1 &= a(1 - e^2) / (1 - e^2 \sin^2 \phi_1)^{3/2} \\
D &= x / (N_1 k_0)
\end{aligned}$$

Figure 2: The Geodetic to UTM Transformation Formulas

One common method of transforming or “reprojecting” a georeferenced image is with a technique called “rubber sheeting.” In rubber sheeting several control points are chosen from both the source and destination space and a linear transformation matrix defined. This technique works well when the transformation can be accomplished with a set of linear equations. In essence, rubber sheeting allows you to rotate, linearly stretch, and translate the source image. Unfortunately, the transformation from UTM

to Geodetic is nonlinear. The result of rubber sheeting on UTM images would be visible discontinuities between adjacent reprojected images as shown in Fig. 3. Where high levels of accuracy are required, rubber sheeting is not a viable method for UTM to Geodetic reprojection.

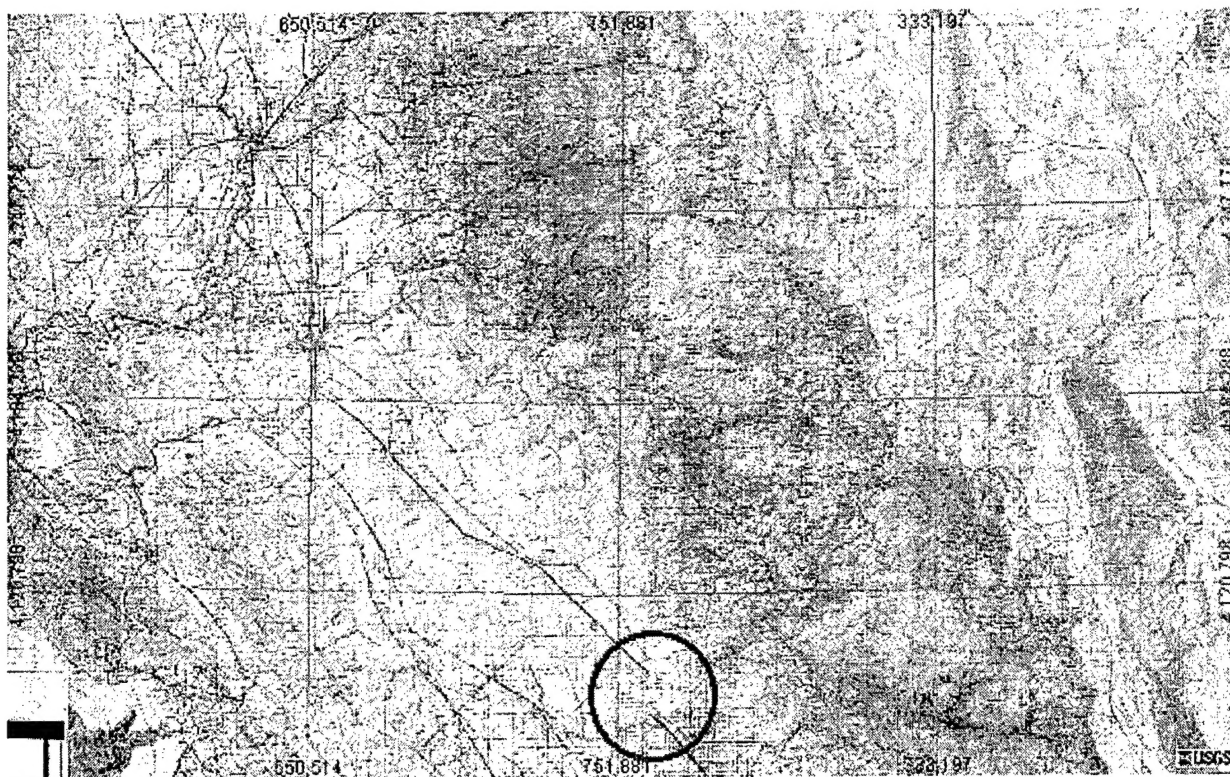


Figure 4: An example of two images reprojected using the rubber sheeting method. Notice the break in the road between the two images (Jain and Barclay, 2003).

POINT WISE REPROJECTION

A projection transformation method exists which is more accurate than the rubber sheeting described above. This process is called point wise reprojection because it operates on each point, or pixel, in the image individually. Instead of transforming the entire image at once this method transforms only one pixel at a time. Transforming one pixel is simple and accurate using the equations given earlier. The process works by determining the color of each pixel in the Geodetic image by converting the pixel's coordinates to UTM and using the color of the corresponding pixel in the UTM image. If the transformation is done for each pixel the result is a Geodetic reprojection of the original UTM image. This method has previously been described by Jain and Barclay (2003), but the following is a more complete explanation of the process.

Point wise transformation transforms the coordinates of each pixel in a Geodetic image into UTM. Initially, an empty Geodetic image must be created to provide the pixels used in the later steps. This

image is made by converting each corner of the UTM image into a Geodetic point. These new Geodetic points form the four corners of a new non-rectangular quadrilateral. Of course, images are rectangles (not misshapen quadrilaterals), so the bounds of the new image will be created from the minimum bounding box of the quadrilateral.

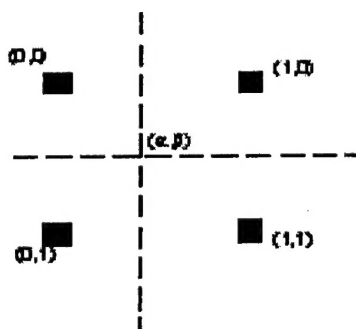


Figure 5: Bilinear Interpolation. The value of the middle point is a weighted average of the bounding points (Jain and Barclay, 2003).

A problem arises when making an empty Geodetic image from the four converted corner points. UTM and Geodetic coordinates are real numbers. Images, on the other hand, are made up of pixels which have integer coordinates. Therefore, the location points must be approximated to integer coordinates whose minimum value is zero in order to map to a specific pixel location in the image. A point to pixel conversion must be done as well as a projection conversion. The dimensions of the bounding Geodetic bounding box (in degrees) divided by the dimensions of the UTM image (in pixels) will give a point per pixel ratio which can be used to determine the dimensions of the empty Geodetic image. A second problem is that the georeferenced coordinates put the origin of an area at the lower left corner whereas digital images, and the programming language which operate on them, specify the origin to be the upper left pixel. Care must be taken to account for this when converting from georeferenced coordinates to pixel coordinates.

Now the actual conversion of the image takes place. If the following procedure is repeated for every pixel in the Geodetic image it will be colored completely, creating a completely reprojected image. First a pixel is chosen in the Geodetic image. The georeferenced coordinates of that pixel are then calculated. If the coordinates are not in the Geodetic quadrilateral defined earlier, the pixel in the Geodetic image are made transparent or a default color. If the Geodetic coordinates are within the quadrilateral the coordinates are reprojected into UTM coordinates. The new UTM coordinates are converted to a pixel location on the UTM image. The pixel of the Geodetic image should be given the same color value as the pixel that was found in the UTM image.

Most likely the UTM coordinates will not convert to a pixel location. It is highly unlikely that the UTM coordinates are even integers. Instead, the UTM point will be located inside a bounding box made up of four pixels in the UTM image. A solution would be to round the newly calculated UTM coordinates to the nearest of these four pixels. That pixel's color would then be given to the Geodetic pixel. In fact, this method performs fairly well, producing Geodetic images which look very similar to the original UTM. But a better method exists.

Instead of setting the Geodetic pixel to the exact value of a UTM pixel, the four closest UTM pixels are averaged together and that color is used as the new value for the Geodetic pixel. The average is actually weighted based on the distance of the new UTM point from each of the four bounding pixels. Fig. 6 shows the original point within its four bounding pixels. This weighted average, commonly known as bilinear interpolation, is performed with the following equation given by Jain and Barclay (2003):

$$f(a,b) = (1-a)(1-b)f(0,0) + a(1-b)f(1,0) + (1-a)b f(0,1) + ab f(1,1)$$

where:

$f(0,0)$ = color of lower left bounding pixel

$f(0,1)$ = color of upper left bounding pixel

$f(1,0)$ = color of lower right bounding pixel

$f(1,1)$ = color of upper right bounding pixel

a = horizontal distance from $f(0,0)$ of the new UTM point (normalized between 0 and 1)

b = vertical distance from $f(0,0)$ of the new UTM point (normalized between 0 and 1)

Usually, each pixel contains a color value encoded as an RGB (red, green, blue) integer. The separate red, green, and blue values are each a color band. The bilinear interpolation must be performed for each separate color band of the pixel. The bands should be recombined after interpolation to give the color of the Geodetic pixel. Bilinear interpolation produces high detail and accurate colors in the reprojected images, noticeably better than just rounding pixels.

REPROJECTION PERFORMANCE

Using the point wise reprojection to convert images from UTM to the Geodetic works well; however, the pixel by pixel approach becomes processor intensive as the size of the images increases. For each pixel there is a conversion of the coordinates from Geodetic to UTM. A 7000x7000 pixel image, the approximate size of the DOQs, requires 49 million point conversions. Each of these point conversions from Geodetic to UTM is non-trivial requiring numerous double precision mathematical operations, many of them trigonometric. Since the entire application was implemented in Java the processing speed became a problem. Performance in Java is a topic for much debate, but double precision math operations are always guaranteed to be slower than compiled languages like C. Java implements the full IEEE requirements for double precision calculations which require those operations to be written without the full CPU optimizations that could be done in C or C++ (which do not follow the IEEE requirements). The result is that all functions in the Java Math class run much slower than the equivalents in C. Because each point conversion from Geodetic to UTM uses functions from the Math class repeatedly (mostly the trigonometric and exponentiation functions), 49 million of these conversions took unacceptably long to complete. Given that thousands of these images are to be converted, each with 49 million pixels, optimization was a necessity.

One optimization technique is to decrease the number of trigonometric functions used in the conversion process. The program code based on the formulas in Figures 1 and 2 had not been completely optimized. First, computations were reused as much as possible. Instead of using the `pow()` (exponentiation) function, simple multiplication was done in its place. Since usually a variable was raised to different powers multiple times, precomputing allowed reusing those variables to save time. Similarly, tangents were replaced with divisions using the already calculated sine and cosine values. Secondly, trigonometric identities replaced trigonometric function calls. This method was especially useful when the identities could be chained to calculate multiple necessary values such as $\sin(n\pi)$ for sequential n . These computational optimizations reduced the total processing time of the system but the performance was still unacceptably slow. The technique described below proved more successful.

PRECOMPUTED TABLE

Optimizing the Geodetic to UTM point transformation did not sufficiently decrease the runtime of image reprojection. The next logical optimization was to decrease the number of point transformations required to reproject an image. Each pixel has to be transformed from Geodetic to UTM in our image transformation algorithm. But instead of calculating each point approximating the transformations for certain points would decrease the total number of calculations required to transform an image. Precomputing a table of Geodetic to UTM points would allow efficient approximation of the point transformations. Prior to processing the image a 300x300 table is filled with conversion values. Each time a pixel of the image is processed a check is made to determine if its coordinates are in the table. If not (the usual case), then the surrounding horizontal and vertical coordinates are used. Then using linear interpolation, their associated UTM coordinates are averaged to give the UTM coordinate of the pixel currently being transformed. With the addition of the table each pixel only requires a simple averaging of points rather than calculations using the complex formulas. A 300x300 table requires only 90,000 point transformations in comparison to the 49 million for the original method.

The linear interpolation used to determine the UTM coordinates of a geodetic point is done separately for the horizontal and vertical components of the point. The basic procedure given the horizontal component of a geodetic point is as follows:

1. Find the surrounding geodetic points in the table.
2. Get the UTM coordinates of those points.
3. Let $x(0,0)$ = the UTM horizontal coordinate of the lower point.
4. Let $x(1,0)$ = the UTM horizontal coordinate of the upper point. Let a = horizontal distance from $x(0,0)$ normalized to $[0, 1]$.
5. Let a = horizontal distance from $x(0,0)$ normalized to $[0, 1]$.
6. $x_{UTM} = (1-a)x(0,0) + ax(1,0)$

The result is the horizontal component of the UTM point. The vertical component can be calculated similarly.

The accuracy of the computation using the table is high. Only 1cm of error is obtained on USGS DOQs with the default 300x300 table. However, if higher accuracy is needed the dimensions of the table can be increased. The table incorporates an accuracy check which determines if the table is within 1m of error for the current image. The DOQs have a 1m per pixel resolution so the error from approximating transformation should be less than ½ m. If it is not then the dimensions of the table will be increased until there is less than 1m of error. Given the current performance of reprojection using the table approximation, the default table would probably give adequately small error for most applications.

Decreasing the dimensions of the table could be considered in order to save time. A reduction in accuracy might be acceptable for some applications if the performance of the reprojection increased. However, creating the transformation table requires a certain amount of processor overhead regardless of the size of the table. The time to create 300x300 default table is not significantly larger than a 100x100 table. A significant reduction in creation time would require a large dimension decrease. Such a drastic change in the size of the table would incur an accuracy loss out of the acceptable range for most applications.

PROCESSING WITH FIDUCIALS

The UTM to geodetic image converter is primarily used to convert digital orthophotos from USGS. These images contain special markers called fiducials whose coordinates are usually given in the header files accompanying the images. Each image has four fiducials. The points marked by the fiducials become the four corners of a rectangle when transformed from the UTM to the Geodetic projections. This rectangle defines the border of the Geodetic image. Normally, the adjacent images contain overlap, but if those images are cropped along the border defined by the fiducials then the overlap is completely removed. If properly converted the images should look continuous but have no overlap after being cropped on the fiducials.

We use the fiducials to decrease the processing time of the images. If adjacent images are being converted then the overlap is unimportant. Instead of converting the images and then cropping on the fiducials we only convert those pixels that are within the fiducial border. This method prevents the redundant conversion. In fact, almost 25% of the image does not have to be converted if we use the fiducials as our initial boundary. When used with the table approximation processing time is decreased substantially.

RESULTS

The reprojection methods described above work well. The transformed images combine with other Geodetic features correctly: roads and rivers in the reprojected images overlap with the Geodetic road and river features. The optimizations using a table and fiducials increase performance speed substantially. Table 1 shows the times for reprojection with the various optimizations. The tests shown in the chart were run on a Pentium III 927Mhz computer with 512MB of RAM. Of course, times would be shorter with a faster computer and more memory.

	No Optimizations	Table	Fiducials	Table and Fiducials
Trial 1	608.17	500.65	477.13	368.80
Trial 2	603.68	501.21	475.88	368.55
Trial 3	607.36	500.72	476.58	370.88
Trial 4	606.74	505.35	475.83	369.5
Trail 5	605.23	501.29	475.66	368.61
Average	606.24	501.84	476.22	369.27
% Decrease	0.00%	17.22%	21.45%	39.09%

Table 1: Runtimes for image reprojections with various optimizations. Times are in seconds.

The following figures show the results of reprojection. The first image, Fig. 7, is an original UTM projected image. The times shown in Table 2 refer to the reprojection of the image in Fig. 8. The transformation does change the image drastically. Fig. 9 shows the results of transformation using table approximation and using the fiducial boundaries. The image is visually very similar to the original. There is noticeable cropping because only the area within the fiducials was transformed. However, the effects of reprojection are barely noticeable. A slight change in the angles of lines, like roads, is probably the most noticeable difference. Of course, little change is desirable; reprojection is not supposed to change the content of the images. Fig. 10 is a good example of the accuracy of this method. This figure shows a section of two adjacent images. Notice the continuity between the roads in the two images. One can compare the accuracy of this image to Fig. 11 where there was a large discontinuity the road that traverses two adjacent images. The ability to preserve continuity between images is the very reason point-wise reprojection was the method of choice for this project.

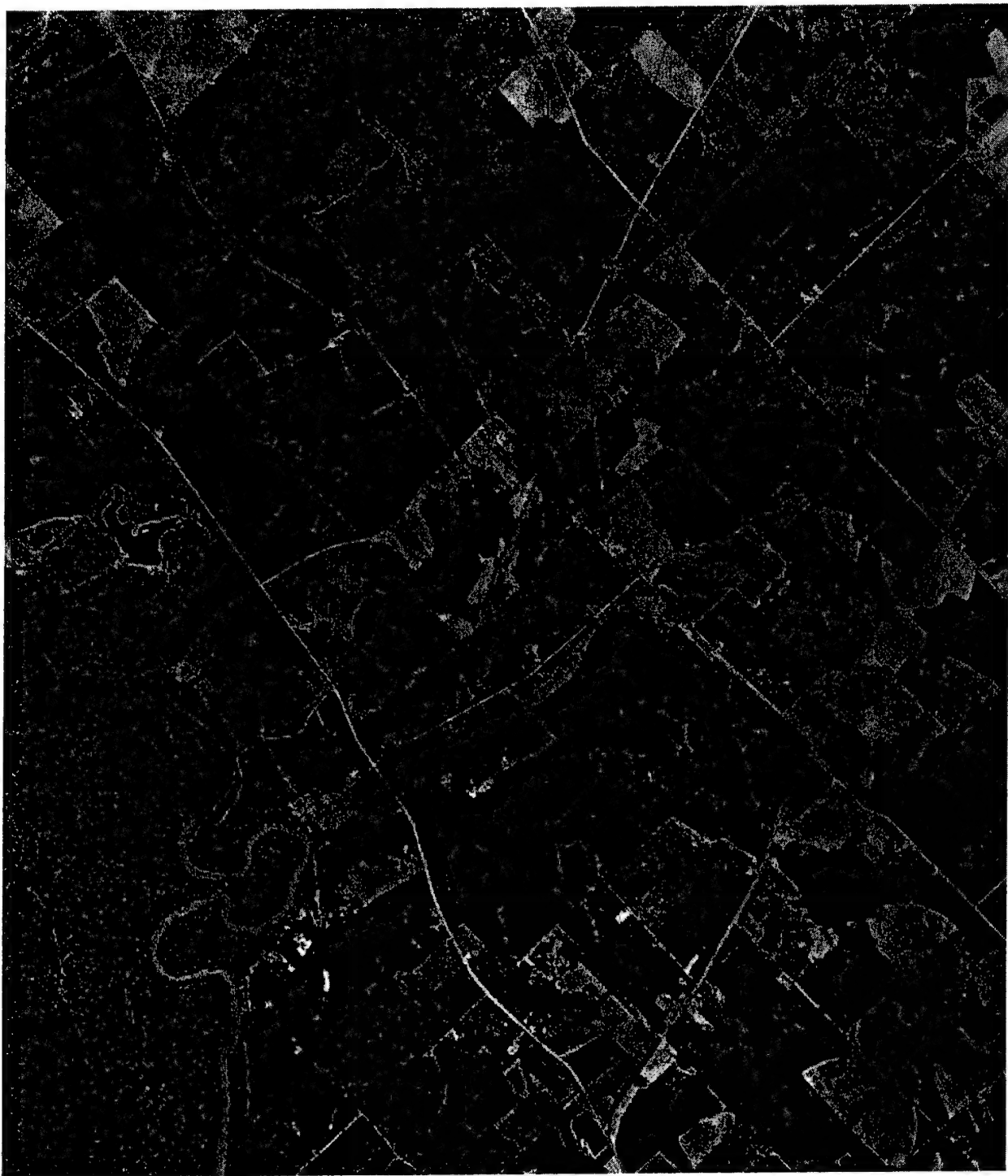


Figure 12: An original UTM DOQ of an area in Georgia

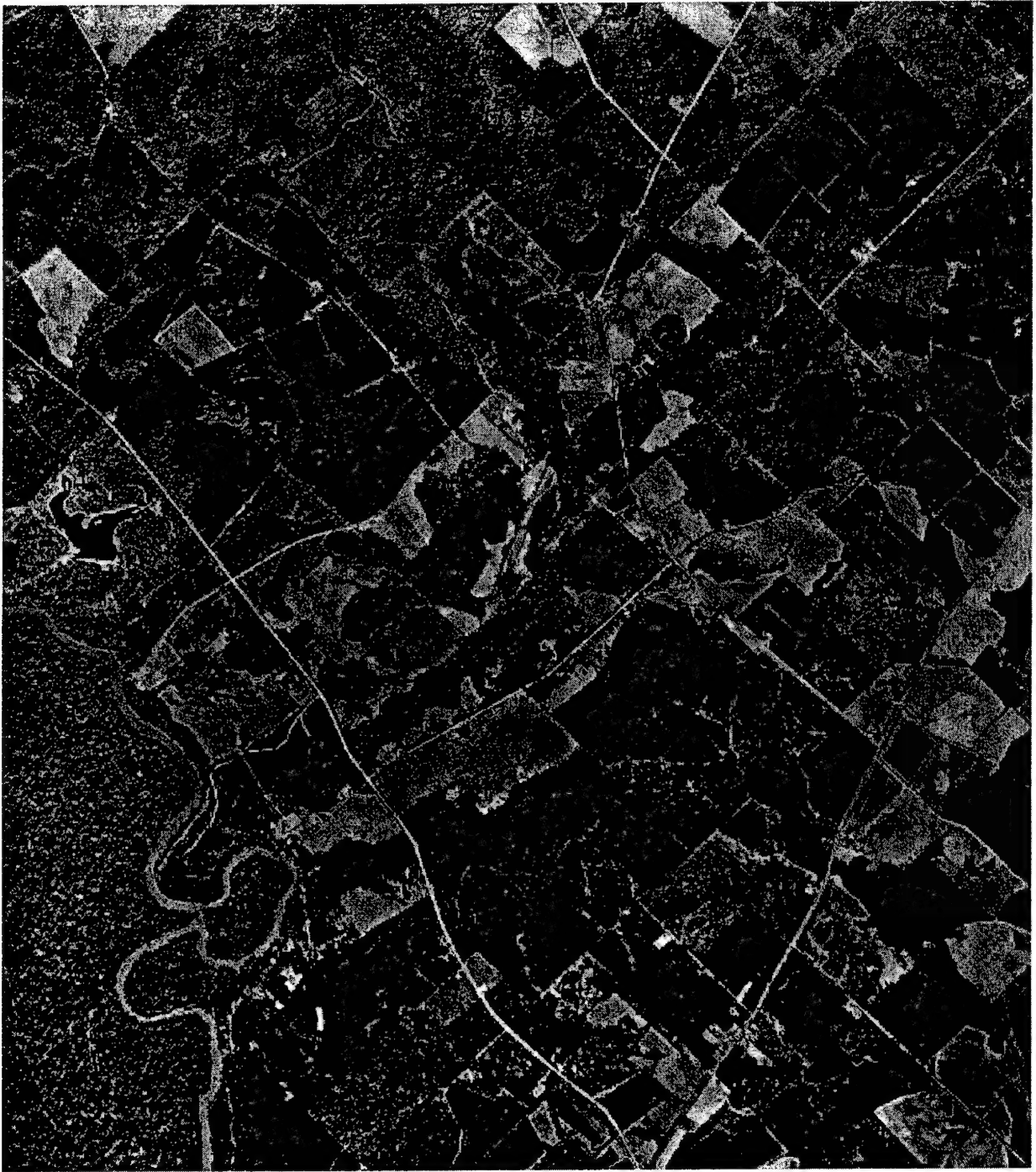


Figure 13: A Geodetic reprojection transformed inside the fiducial boundary using the table approximation.

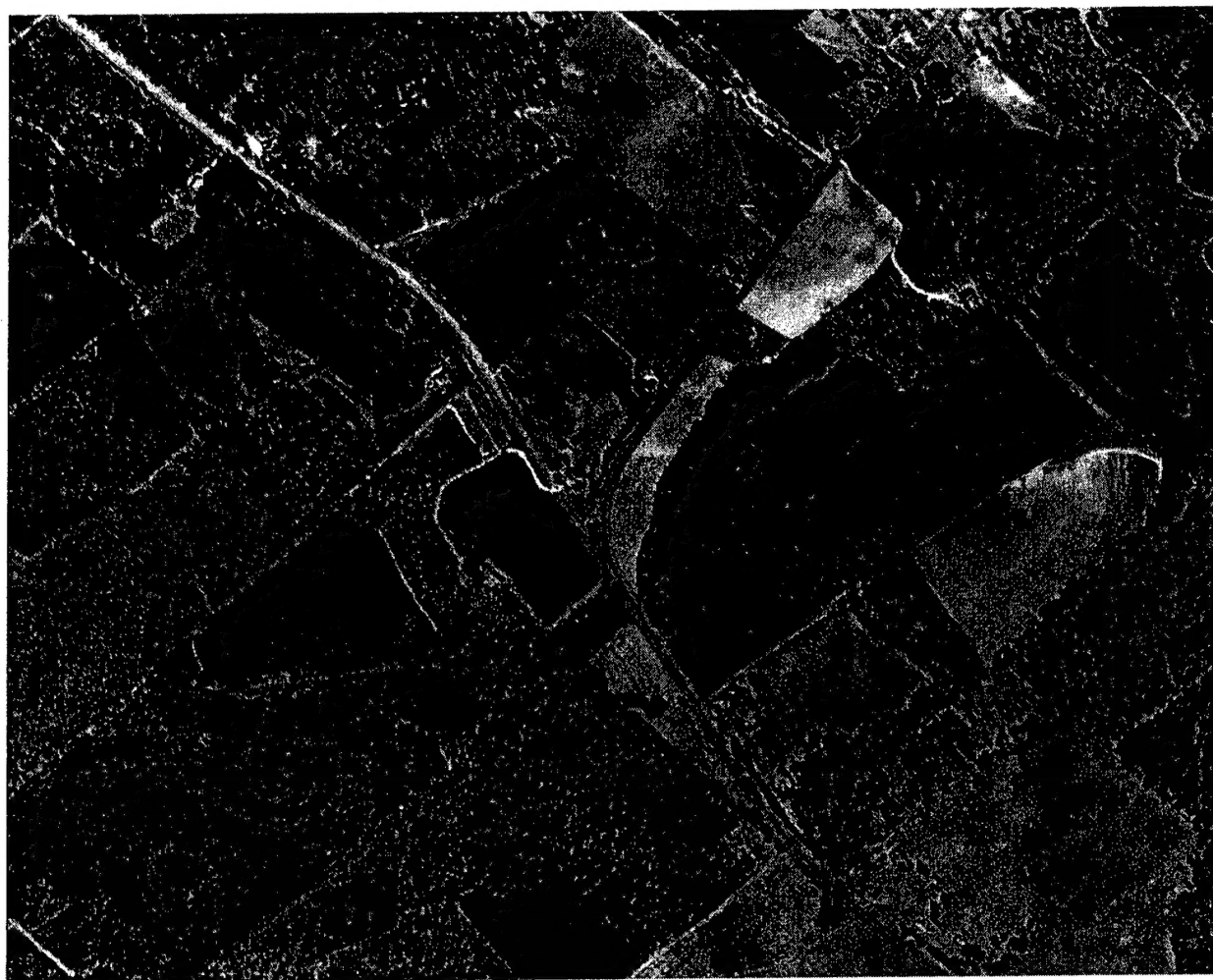


Figure 14: The boundary of two converted images. Notice the continuity of the roads.

CONCLUSIONS

The optimized image transformation program has successfully converted all 4000 DOQs of the state of Georgia. The 4000 images were all transformed in a matter of days and are currently in use in GIS systems which use the Geodetic projection. The results match with those already in the system and have provided a valuable new geospatial resource. Despite the fact that there were tremendous speed increases over the course of the project, even the fastest runs took minutes to complete. These speeds preclude using the software to reproject images on demand, a major limitation. One simple solution would be to reimplement the procedure in a compiled programming language like C. Performance would increase substantially and would likely allow for on-demand processing on a fast server. Portability would be the main loss in this situation, since at a minimum each architecture would require a different natively compiled solution. Ease of integration with existing components would also be lost. Most of the related software that would use this tool is written in Java, thus a Java solution to this problem is desirable.

Of course, the current Java implementation might be suitable for online applications if the image sizes are small. A 49 million pixel image is rather large and often not suitable for online processing. If the image is reduced to a manageable size and a fast computer is used, the process could be quite successful in online applications.

The method described in this paper is not specific to UTM to Geodetic transformations. It could be done with any two projection for which accurate point transformations can be made. This includes any combinations of the following projections: Transverse Mercator, Lambert Polyconic, Polar Steriographic, Equidistant Azimuthal, etc.

ACKNOWLEDGEMENTS

We would like to thank the National Guard Bureau – Counterdrug for sponsoring this work and providing imagery for the state of Georgia used in this project.

REFERENCES

- Jain, Siddharth and Tom Barclay. *Adding the EPSG:4326 Geographic Longitude-Latitude Projection to TerraServer*. Technical Report MSR-TR-2003-56. Microsoft Research, Redmond: 2003.
- Snyder, John P. *Map Projections – A Working Manual*. U.S. Geological Survey Professional Paper 1395. United States Government Printing Office, Washington: 1987.
- U.S. Geological Survey, 1996, *Standards for Digital Orthophotos: Specifications*.